



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/671,889	09/29/2003	Fred Gehrung Gustavson	Y0R920030170US1	8009
48150	7590	11/06/2008	EXAMINER	
MCGINN INTELLECTUAL PROPERTY LAW GROUP, PLLC			VICARY, KEITH E	
8321 OLD COURTHOUSE ROAD			ART UNIT	PAPER NUMBER
SUITE 200				2183
VIENNA, VA 22182-3817			MAIL DATE	DELIVERY MODE
			11/06/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/671,889	Applicant(s) GUSTAVSON ET AL.
	Examiner Keith Vicary	Art Unit 2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If no period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED. (35 U.S.C. § 133).

Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 20 August 2008.

2a) This action is FINAL. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-9 and 11-19 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-9 and 11-19 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All b) Some * c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO-166/08)
Paper No(s)/Mail Date 8/29/2008

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____

5) Notice of Informal Patent Application

6) Other: _____

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

0. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 8/29/2008 has been entered.

1. Claims 1-9 and 11-19 are pending in this office action and presented for examination. Claims 1, 6, 12, and 17 are newly amended by amendment filed 8/20/2008.

Specification

2. The amendment filed 8/20/2008 is objected to under 35 U.S.C. 132(a) because it introduces new matter into the disclosure. 35 U.S.C. 132(a) states that no amendment shall introduce new matter into the disclosure of the invention. The added material which is not supported by the original disclosure is as follows.

Applicant is required to cancel the new matter in the reply to this Office Action.

3. In the first paragraph added to the specification, an explicit definition for "non-standard format" is given. Although the '888 application does disclose that a permutation does not lead to a standard row or column major representation, the '888

application does not disclose of a "non-standard format" nor give an explicit definition for it.

4. The second paragraph added to the specification discloses that $k > 1$ indicates a number of data capable of being simultaneously moved in a single instruction, which is of a different scope than the '888 application's apparent teaching that $k > 1$ indicates a machine has multiple SIMD FPUs.

5. The third paragraph added to the specification discloses that "associated floating point registers" are abbreviated to be FPUs instead of FRegs, and also discloses that different architectural or instruction set scenarios would "require" a need to lay out the blocks differently instead of "provid[ing]" a need to lay out the blocks differently as disclosed in the co-pending application.

Claim Objections

6. Claim 1 is objected to because of the following informalities. Appropriate correction is required.

a. Claim 1, line 9 recites the limitation "before it is scheduled to be used" which presumably should be "before it is scheduled to be used."

Double Patenting

7. Claims 1-9 and 11-19 of this application conflict with claims 1, 3-6, 8-12, and 14-19 of Application No. 10671937. 37 CFR 1.78(b) provides that when two or more

applications filed by the same applicant contain conflicting claims, elimination of such claims from all but one application may be required in the absence of good and sufficient reason for their retention during pendency in more than one application.

Applicant is required to either cancel the conflicting claims from all but one application or maintain a clear line of demarcation between the applications. See MPEP § 822.

8. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

9. Claims 1-9 and 11-19 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1, 3-6, 8-12, and 14-19 of copending Application No. 10671937 in view of Gustavson et al. (Gustavson) (Superscalar GEMM-based Level 3 BLAS – The On-going Evolution of a Portable and High-Performance Library, Para'98, pages 207-215). Although the conflicting claims

are not identical, they are not patentably distinct from each other because claims 1-9 and 11-19 of the instant application are obvious variants of claims 1, 3-6, 8-12, and 14-19 of the '937 application.

This is a provisional obviousness-type double patenting rejection.

10. Claims 1-9 and 11-19 of the instant application contain every limitation of claims 1, 3-6, 8-12, and 14-19 of the '937 application; moreover, claims 1-9 and 11-19 of the instant application claim disclose inserting instructions to move data into said cache providing data into an FPU so that said LSUs can move said data into said Fregs in a timely manner for said linear algebra subroutine execution, whereas claims 1, 3-6, 8-12, and 14-19 of the '937 application merely claim preloading data into a floating point register of an FPU. Moreover, claims 1-9 and 11-19 of the instant application also disclose of data being prefetched into said cache from a memory in a nonstandard format predetermined to reduce a number of data streams for a level 3 processing to be three streams and to allow a multiple loading of loads into said FPU by said LSU.

First, it would have been readily recognized by one of ordinary skill in the art at the time of the invention that the benefits of using cache in the '937 application are numerous and include greater system performance due to the decreased access time to access cache in comparison to main memory combined with the locality of reference that is typical in most computer programs.

It would have been obvious to one of ordinary skill in the art at the time of the invention to implement cache into the '937 application to gain greater system

performance; it would have been readily recognized by one of ordinary skill in the art at the time of the invention that greater system performance is desirable in any processor. Furthermore, it would have been readily recognized by one of ordinary skill in the art at the time of the invention that this cache would fit into the '937 application by receiving data from the main memory and sending it to the floating point register, and that when preloading data into the floating point register in a system which uses a cache, that data would have to be prefetched into the cache in order to be preloaded into the register.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the widely-known teachings of cache with the invention of the '937 application in order to increase system performance.

Moreover, claims 1-9 and 11-19 of the instant application also disclose of data being prefetched into said cache from a memory in a nonstandard format predetermined to reduce a number of data streams for a level 3 processing to be three streams and to allow a multiple loading of loads into said FPU by said LSU.

On the other hand, Gustavson discloses, said data being prefetched into said cache from a memory in a nonstandard format (section 3.1, first indented paragraph of page 210, technique of keeping a small square block of C in registers; this technique of prefetching C in the format of a small square block as opposed to the prefetching of A and B can be considered nonstandard) to reduce a number of data streams for a level 3 processing to be three streams (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; note that as only a small square block of C instead of the entire C is being loaded into the registers, C is

essentially a data stream of small square blocks. Also note that streams can be broadly read to be the data from the FPU registers to the FPU itself and thus encompasses A, B, and C regardless of the above technique) and to allow a multiple loading of loads into said FPU by said LSU (section 3.1, first indented paragraph of page 210 as above, number of load and store instructions; there thus must exist multiple loads into said FPU by said LSU).

Gustavson's teaching above maximizes the ratio between the number of MAAs and the number of load and store instructions used to transfer data to and from registers (section 3.1, page 210, first indented paragraph, first 5 lines).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Gustavson with the invention of the '937 application in order to maximize the ratio between the number of MAAs and the number of load and store instructions, which enables the increase in system performance. It would have been readily recognized to one of ordinary skill in the art at the time of the invention that the teaching of Gustavson does not render the invention of the '937 application unusable. The claims of the '937 application disclose of preloading data to an FPU for linear algebra operations so that the data may be timely executed by the FPU but does not disclose the format of the data or the format of how the preloading is actually done. Gustavson teaches the above limitations in describing how to gain an increase in system performance when executing linear algebra operations.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Gustavson with the invention of the '937

application in order to maximize the ratio between the number of MAAs and the number of load and store instructions, which enables the increase in system performance

b. Further note that claims 2, 11, and 13 in the instant application also claim that prefetching data is accomplished by utilizing time slots caused by a difference between a time to execute instructions in said subroutine execution process and a time to load said data, while claims 1, 11, and 12 of the '937 application does not explicitly disclose this.

It would have been readily recognized by one of ordinary skill in the art at the time of the invention that prefetching data in general cuts down the amount of time a processor is waiting for a memory miss to be serviced, and prefetching by utilizing time slots caused by a difference between a time to execute instructions and a time to load said data allows for data to be prefetched ahead of time without delaying any other instructions that are being processed. Furthermore, it would have been readily recognized by one of ordinary skill in the art at the time of the invention that the benefits of prefetching are contingent upon other instructions not being delayed due to the prefetching; thus, it would have been readily recognized to one of ordinary skill in the art at the time of the invention that prefetching would be done by utilizing these time slots of inactivity.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to combine the widely-known method of prefetching by utilizing time slots with the '937 application in order to cut down the amount of

time a processor is waiting for a memory miss to be serviced, thus increasing overall system performance.

11. Aside from the obvious variants listed above, claim 1 of the '937 application contains every element of claim 1 of the instant application.
12. Aside from the obvious variants listed above, claim 1 of the '937 application contains every element of claim 2 of the instant application.
13. Aside from the obvious variants listed above, claim 3 of the '937 application contains every element of claim 3 of the instant application.
14. Aside from the obvious variants listed above, claim 4 of the '937 application contains every element of claim 4 of the instant application.
15. Aside from the obvious variants listed above, claim 5 of the '937 application contains every element of claim 5 of the instant application.
16. Aside from the obvious variants listed above, claim 6 of the '937 application contains every element of claim 6 of the instant application.
17. Aside from the obvious variants listed above, claim 8 of the '937 application contains every element of claim 7 of the instant application.
18. Aside from the obvious variants listed above, claim 9 of the '937 application contains every element of claim 8 of the instant application.
19. Aside from the obvious variants listed above, claim 10 of the '937 application contains every element of claim 9 of the instant application.
20. Aside from the obvious variants listed above, claim 6 of the '937 application contains every element of claim 11 of the instant application.

21. Aside from the obvious variants listed above, claim 12 of the '937 application contains every element of claim 12 of the instant application.
22. Aside from the obvious variants listed above, claim 12 of the '937 application contains every element of claim 13 of the instant application.
23. Aside from the obvious variants listed above, claim 14 of the '937 application contains every element of claim 14 of the instant application.
24. Aside from the obvious variants listed above, claim 15 of the '937 application contains every element of claim 15 of the instant application.
25. Aside from the obvious variants listed above, claim 16 of the '937 application contains every element of claim 16 of the instant application.
26. Aside from the obvious variants listed above, claim 17 of the '937 application contains every element of claim 17 of the instant application.
27. Aside from the obvious variants listed above, claim 18 of the '937 application contains every element of claim 18 of the instant application.
28. Aside from the obvious variants listed above, claim 19 of the '937 application contains every element of claim 19 of the instant application.

Claim Rejections - 35 USC § 112

29. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

30. Claims 1-9 and 11-19 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

31. Claim 1 recites the limitation "LSUs can load said data into said Fregs before it is scheduled to be used in said linear algebra subroutine execution" in line 9. The original disclosure does not disclose the broad interpretation of the claim in which the data is loaded into said Fregs before the instruction which uses said data is scheduled for execution. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

32. Claim 1 recites the limitation "said three data streams comprise data of one matrix...and data for two remaining matrix operands..." in the last 3 lines. The original disclosure does not disclose the broad interpretation of the claim in which each data stream contains data of all three matrixes. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

33. Claim 1 recite the limitation "a nonstandard format predetermined to reduce a number of data streams for a level 3 nested loop matrix-matrix type kernel type operation processing to be three streams" in lines 8-9. This limitation does not appear

to be present in the original disclosure of the instant application. If the limitation is present somewhere in one of the co-pending applications, this should be noted in any subsequent arguments to overcome the rejection. Applicant has previously argued that the present application supports the claim language of reducing the number of data streams to be three streams via various citations. However, the citations given do not appear to support the claim language of *reducing* the number of data streams to be three streams.

- c. Claims 2-5 are rejected for inheriting the defects of base claim 1.

- 34. Claim 6 recites the limitation "said three data streams comprise data of one matrix...and data for two remaining matrix operands..." in the last 4 lines. The original disclosure does not disclose the broad interpretation of the claim in which each data stream contains data of all three matrixes. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

- 35. Claim 6 recite the limitation "a nonstandard format predetermined to reduce a number of data streams for a level 3 linear algebra processing to be three streams" in lines 10-11. This limitation does not appear to be present in the original disclosure of the instant application. If the limitation is present somewhere in one of the co-pending applications, this should be noted in any subsequent arguments to overcome the rejection. Applicant has previously argued that the present application supports the claim language of reducing the number of data streams to be three streams via various

citations. However, the citations given do not appear to support the claim language of *reducing the number of data streams to be three streams*.

d. Claims 7-9 and 11 are rejected for inheriting the defects of base claim 6.

36. Claim 12 recites the limitation "inserting instructions to move data into said cache providing said data into said FPU before it was scheduled to be used for processing in said linear algebra subroutine" in lines 8-10. The original disclosure does not disclose the broad interpretation of the claim in which the data is loaded into said Fregs before the instruction which uses said data is scheduled for execution. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

37. Claim 12 recites the limitation "said three data streams comprise data of one matrix...and data for two remaining matrix operands..." in the last 4 lines. The original disclosure does not disclose the broad interpretation of the claim in which each data stream contains data of all three matrixes. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

38. Claim 12 recites the limitation "a nonstandard format predetermined to reduce a number of data streams for a level 3 linear algebra processing to be three streams" in lines 11-13. This limitation does not appear to be present in the original disclosure of the instant application. If the limitation is present somewhere in one of the co-pending applications, this should be noted in any subsequent arguments to overcome the

rejection. Applicant has previously argued that the present application supports the claim language of reducing the number of data streams to be three streams via various citations. However, the citations given do not appear to support the claim language of *reducing* the number of data streams to be three streams.

e. Claims 13-16 are rejected for inheriting the defects of base claim 12.

39. Claim 17 recites the limitation "inserting instructions to move data into said cache providing said data into said FPU before it was scheduled to be used for processing in said linear algebra subroutine" in lines 6-7. The original disclosure does not disclose the broad interpretation of the claim in which the data is loaded into said Fregs before the instruction which uses said data is scheduled for execution. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

40. Claim 17 recites the limitation "said three data streams comprise data of one matrix...and data for two remaining matrix operands..." in lines 14-17. The original disclosure does not disclose the broad interpretation of the claim in which each data stream contains data of all three matrixes. An amendment to overcome the corresponding indefinite rejection in a manner consistent with the original specification would most likely overcome this rejection as well.

41. Claim 17 recites the limitation "a nonstandard format predetermined to reduce a number of data streams for a level 3 processing to be three streams" in lines 8-9. This

limitation does not appear to be present in the original disclosure of the instant application. If the limitation is present somewhere in one of the co-pending applications, this should be noted in any subsequent arguments to overcome the rejection. Applicant has previously argued that the present application supports the claim language of reducing the number of data streams to be three streams via various citations. However, the citations given do not appear to support the claim language of *reducing* the number of data streams to be three streams.

f. Claims 18-19 are rejected for inheriting the defects of base claim 17.

42. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

43. Claims 1-9 and 11-19 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

44. Claim 1 recites the limitation "LSUs can load said data into said Fregs before it is scheduled to be used in said linear algebra subroutine execution" in line 9. It is indefinite as to whether the data is loaded into said Fregs before the instruction which uses said data is executed, or whether the data is loaded into said Fregs before the instruction which uses said data is scheduled for execution, which occurs beforehand.

45. Claim 1 recites the limitation "p and q are small integers" in line 15. It is indefinite as to what are "small" integers as whether an integer is small or not depends on what it is relative to.

46. Claim 1 recites the limitation "the pieces of these blocks" in line 15. There is insufficient antecedent basis for this limitation in the claim.

47. Claim 1 recites the limitation "said three data streams comprise data of one matrix...and data for two remaining matrix operands..." in the last 3 lines. It is indefinite as to whether one data stream consists of only data of one matrix resident in said cache and the other two data streams each contains data for a respective remaining matrix operand of the two matrix operands, or whether each data stream contains data of all three matrixes.

48. Claim 1 recites the limitation "said level 3 processing" in line 17 and 19. There is insufficient antecedent basis for this limitation in the claim.

g. Claims 2-5 are rejected for failing to alleviate the rejection of claim 1 above.

49. Claim 2 recites the limitation "said timely moving data" in line 1. There is insufficient antecedent basis for this limitation in the claim.

50. Claim 6 recites the limitation "p and q are small integers" in line 14. It is indefinite as to what are "small" integers as whether an integer is small or not depends on what it is relative to.

51. Claim 6 recites the limitation "the pieces of these blocks" in line 14-15. There is insufficient antecedent basis for this limitation in the claim.
52. Claim 6 recites the limitation "said three data streams comprise data of one matrix...and data for two remaining matrix operands..." in the last 4 lines. It is indefinite as to whether one data stream consists of only data of one matrix resident in said cache and the other two data streams each contains data for a respective remaining matrix operand of the two matrix operands, or whether each data stream contains data of all three matrixes.
53. Claim 6 recite the limitation "level 3 linear algebra processing" in, for example, line 11 of claim 6. It is indefinite as to what exactly a "level 3 linear algebra processing" is. Applicant argues that "Level 3 processing" is a commonly-used term by the DLA community to mean doing $O(n^3)$ operations on $O(n^2)$ data. However, page 12 of the instant specification discloses that the limitation "Level 3" means that the kernel involves three loops. Note that this definition does not necessarily mean that the loops are nested. Therefore, it is indefinite as to whether the aforementioned limitation means that the kernel involves three loops, or doing $O(n^3)$ operations on $O(n^2)$ data.
54. Claim 6 recites the limitation "a nonstandard format...wherein said nonstandard format comprises a register block format" in lines 10-13. It is indefinite as to what a nonstandard format is. The meaning of a "non-standard format" even to people in the art may change over time and thus the limitation is indefinite. Applicant states that description of the standard data format is present on various co-pending applications that have been incorporated by reference. However, these co-pending applications

disclose that of "the standard column major format of A." There remains no explicit definition of the limitation "standard format" or "non-standard format." Applicant additionally cites lines 12-15 of page 12; however, this citation likewise does not explicitly define the aforementioned limitations. Even if a portion of the specification "hints" as to the meaning of a certain limitation in the claims, this by itself does not necessarily make the limitation definite. It is further unclear as to how it is implicit that the matrix is stored in one of the two standard formats of DLA. Applicant again cites in the top of page 13 of two co-pending applications which describe non-standard data structures. However, these data structures are not explicitly defined to be non-standard data structures. Even though these data structures may be considered "non-standard data structures," this does not mean that the limitation "nonstandard format" cannot be read broadly as in the rejection.

Moreover, because the limitation "non-standard format" is not explicitly defined in this or any of the co-pending applications, the limitation is still taken to be new matter. Although a specific format (the species) which may be considered as non-standard may be described in the co-pending applications, the claimed genus of a "non-standard format" does not seem to be supported by the instant and any co-pending applications. Because the claimed invention of the instant and co-pending applications seems directed toward specific non-standard formats, and not any non-standard format, the new matter rejection is maintained.

Applicant's amended limitation discloses that "said nonstandard format comprises a register block format." However, it appears that a nonstandard format is

not synonymous with a register block format, as a register block format includes a block format wherein blocks are laid out either in row- or column-major format, which would be able to be considered a standard format. Therefore, since it appears that a nonstandard format is considered nonstandard due to other reasons besides whether the format is a register data block data format, it remains indefinite as to what makes a format nonstandard. Examiner preliminarily recommends replacing the "non-standard" language with language specifying that a block is of a format different than that of a row- or column-major format.

h. Claims 7-9 and 11 are rejected for failing to alleviate the rejection of claim 6 above.

55. Claim 12 recites the limitation "inserting instructions to move data into said cache providing said data into said FPU before it was scheduled to be used for processing in said linear algebra subroutine" in line 7-10. It is indefinite as to whether the data is moved before the instruction which uses said data is executed, or whether the data is moved before the instruction which uses said data is scheduled for execution, which occurs beforehand. It is indefinite as to whether it is the moving of data into said cache or the providing data into said FPU which is done before it was scheduled to be used for processing in said linear algebra subroutine.

56. Claim 12 recites the limitation "p and q are small integers" in line 16. It is indefinite as to what are "small" integers as whether an integer is small or not depends on what it is relative to.

57. Claim 12 recites the limitation "the pieces of these blocks" in line 16-17. There is insufficient antecedent basis for this limitation in the claim.
58. Claim 12 recites the limitation "said three data streams comprise data of one matrix...and data for two remaining matrix operands..." in the last 5 lines. It is indefinite as to whether one data stream consists of only data of one matrix resident in said cache and the other two data streams each contains data for a respective remaining matrix operand of the two matrix operands, or whether each data stream contains data of all three matrixes.
59. Claim 12 recite the limitation "level 3 linear algebra processing" in, for example, line 12-13. It is indefinite as to what exactly a "level 3 linear algebra processing" is. Applicant argues that "Level 3 processing" is a commonly-used term by the DLA community to mean doing $O(n^3)$ operations on $O(n^2)$ data. However, page 12 of the instant specification discloses that the limitation "Level 3" means that the kernel involves three loops. Note that this definition does not necessarily mean that the loops are nested. Therefore, it is indefinite as to whether the aforementioned limitation means that the kernel involves three loops, or doing $O(n^3)$ operations on $O(n^2)$ data.
60. Claim 12 recites the limitation "a nonstandard format...wherein said nonstandard format comprises a register block format" in lines 11-15. It is indefinite as to what a nonstandard format is. The meaning of a "non-standard format" even to people in the art may change over time and thus the limitation is indefinite. Applicant states that description of the standard data format is present on various co-pending applications that have been incorporated by reference. However, these co-pending applications

disclose that of "the standard column major format of A." There remains no explicit definition of the limitation "standard format" or "non-standard format." Applicant additionally cites lines 12-15 of page 12; however, this citation likewise does not explicitly define the aforementioned limitations. Even if a portion of the specification "hints" as to the meaning of a certain limitation in the claims, this by itself does not necessarily make the limitation definite. It is further unclear as to how it is implicit that the matrix is stored in one of the two standard formats of DLA. Applicant again cites in the top of page 13 of two co-pending applications which describe non-standard data structures. However, these data structures are not explicitly defined to be non-standard data structures. Even though these data structures may be considered "non-standard data structures," this does not mean that the limitation "nonstandard format" cannot be read broadly as in the rejection.

Moreover, because the limitation "non-standard format" is not explicitly defined in this or any of the co-pending applications, the limitation is still taken to be new matter. Although a specific format (the species) which may be considered as non-standard may be described in the co-pending applications, the claimed genus of a "non-standard format" does not seem to be supported by the instant and any co-pending applications. Because the claimed invention of the instant and co-pending applications seems directed toward specific non-standard formats, and not any non-standard format, the new matter rejection is maintained.

Applicant's amended limitation discloses that "said nonstandard format comprises a register block format." However, it appears that a nonstandard format is

not synonymous with a register block format, as a register block format includes a block format wherein blocks are laid out either in row- or column-major format, which would be able to be considered a standard format. Therefore, since it appears that a nonstandard format is considered nonstandard due to other reasons besides whether the format is a register data block data format, it remains indefinite as to what makes a format nonstandard. Examiner preliminarily recommends replacing the "non-standard" language with language specifying that a block is of a format different than that of a row- or column-major format.

- i. Claims 13-16 are rejected for failing to alleviate the rejection of claim 12 above.
61. Claim 13 recites the limitation "said timely moving data" in line 1. There is insufficient antecedent basis for this limitation in the claim.
62. Claim 17 recites the limitation "instructions are inserted to move data into a cache providing data to said FPU before it is scheduled to be used in the linear algebra subroutine" in lines 6-7. It is indefinite as to whether the data is moved before the instruction which uses said data is executed, or whether the data is moved before the instruction which uses said data is scheduled for execution, which occurs beforehand. It is indefinite as to whether it is the moving of data into said cache or the providing data into said FPU which is done before it was scheduled to be used for processing in said linear algebra subroutine.

63. Claim 17 recites the limitation "p and q are small integers" in line 12. It is indefinite as to what are "small" integers as whether an integer is small or not depends on what it is relative to.

64. Claim 17 recites the limitation "the pieces of these blocks" in lines 12-13. There is insufficient antecedent basis for this limitation in the claim.

65. Claim 17 recites the limitation "said three data streams comprise data of one matrix...and data for two remaining matrix operands..." in lines 14-17. It is indefinite as to whether one data stream consists of only data of one matrix resident in said cache and the other two data streams each contains data for a respective remaining matrix operand of the two matrix operands, or whether each data stream contains data of all three matrixes.

66. Claim 17 recites the limitation "a nonstandard format...wherein said nonstandard format comprises a register block format" in lines 8-13. It is indefinite as to what a nonstandard format is. The meaning of a "non-standard format" even to people in the art may change over time and thus the limitation is indefinite. Applicant states that description of the standard data format is present on various co-pending applications that have been incorporated by reference. However, these co-pending applications disclose that of "the standard column major format of A." There remains no explicit definition of the limitation "standard format" or "non-standard format." Applicant additionally cites lines 12-15 of page 12; however, this citation likewise does not explicitly define the aforementioned limitations. Even if a portion of the specification "hints" as to the meaning of a certain limitation in the claims, this by itself does not

necessarily make the limitation definite. It is further unclear as to how it is implicit that the matrix is stored in one of the two standard formats of DLA. Applicant again cites in the top of page 13 of two co-pending applications which describe non-standard data structures. However, these data structures are not explicitly defined to be non-standard data structures. Even though these data structures may be considered "non-standard data structures," this does not mean that the limitation "nonstandard format" cannot be read broadly as in the rejection.

Moreover, because the limitation "non-standard format" is not explicitly defined in this or any of the co-pending applications, the limitation is still taken to be new matter. Although a specific format (the species) which may be considered as non-standard may be described in the co-pending applications, the claimed genus of a "non-standard format" does not seem to be supported by the instant and any co-pending applications. Because the claimed invention of the instant and co-pending applications seems directed toward specific non-standard formats, and not any non-standard format, the new matter rejection is maintained.

Applicant's amended limitation discloses that "said nonstandard format comprises a register block format." However, it appears that a nonstandard format is not synonymous with a register block format, as a register block format includes a block format wherein blocks are laid out either in row- or column-major format, which would be able to be considered a standard format. Therefore, since it appears that a nonstandard format is considered nonstandard due to other reasons besides whether the format is a register data block data format, it remains indefinite as to what makes a

format nonstandard. Examiner preliminarily recommends replacing the "non-standard" language with language specifying that a block is of a format different than that of a row- or column-major format.

67. Claims 17 recites the limitation "level 3 processing" in, for example, line 9 of claim 17. It is indefinite as to what exactly a "level 3 processing" is. Applicant argues that "Level 3 processing" is a commonly-used term by the DLA community to mean doing $O(n^3)$ operations on $O(n^2)$ data. However, it appears as though "level 3 processing" can also be interpreted as matrix-matrix operations. Although matrix-matrix operations may entail doing $O(n^3)$ operations on $O(n^2)$ data, it is readily recognized that there are other cases where $O(n^3)$ operations are done on $O(n^2)$ data that are unrelated to matrix-matrix operations. Therefore, it is indefinite as to whether Level 3 processing means doing $O(n^3)$ operations on $O(n^2)$ data or doing matrix-matrix operations, as the former may be distinct from the latter. Moreover, page 12 of the instant specification discloses that the limitation "Level 3" means that the kernel involves three loops. Note that this definition does not necessarily mean that the loops are nested. Therefore, it is also indefinite as to whether the aforementioned limitation means that the kernel involves three loops, or doing $O(n^3)$ operations on $O(n^2)$ data.

j. Claims 18-19 are rejected for failing to alleviate the rejections of claim 17 above.

Claim Rejections - 35 USC § 102

68. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

69. Claims 1-9 and 11-19 are rejected under 35 U.S.C. 102(b) as being anticipated by Gustavson et al. (Gustavson) (Superscalar GEMM-based Level 3 BLAS – The Ongoing Evolution of a Portable and High-Performance Library, Para'98, pages 207-215).

70. Consider claim 1, Gustavson discloses for an execution code (section 1, line 6, BLAS code) controlling an operation of said floating point unit (FPU) (section 3.1, line 4, discloses floating point registers, therefore it is inherent there are floating point units that are doing the multiplications as in section 1, line 2) performing a linear algebra subroutine execution (section 1, line 8, routine along with section 1, line 1, linear algebra), inserting instructions to move data in a contiguous and stride one format (page 210, first indented paragraph, discloses of using regular load and store instruction to transfer data to and from registers; a load instruction loads contiguous data at an aligned memory address. Alternatively, section 4 describes of a PowerPC604 which performs loads to access data in a contiguous and stride one format) into a cache providing data for said FPU for direct loading into said FPU (the L1 cache and registers are directly connected), so that said LSUs can load said data into said Fregs before it is scheduled to be used in said linear algebra subroutine execution (section 4.1, line 8, algorithmic prefetching), said data being prefetched into said cache from a memory in a

register block format (the prefetching is described above in section 4.1, see below for the register block format explanations) to reduce a number of data streams for a level 3 nested loop matrix-matrix type kernel type operation processing to be three streams (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; note that as only a small square block of C instead of the entire C is being loaded into the registers, C is essentially a data stream of small square blocks. Also note that streams can be broadly read to be the data from the FPU registers to the FPU itself and thus encompasses A, B, and C regardless of the above technique) and to allow a loading of these streams into said FPU by said LSU (section 3.1, first indented paragraph of page 210 as above, number of load and store instructions), said register block format comprising a data storage format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said FRegs (consider a subset or set of matrix data stored in any format in a memory. That matrix data can be arbitrarily split up into blocks of size p-by-q. Regardless of how small or big these blocks of matrix data are, and what data is within these blocks, single or multiple elements of this block of matrix data can be fitted in some way into said FRegs as is necessary for calculations to be subsequently performed), and wherein said three data streams comprise data of one matrix of said level 3 processing is considered to be resident in said cache and data for two remaining matrix operands of said level 3 processing as residing in a memory or a cache level higher than said cache (section 3.1, first indented paragraph of page 210 as above,

three total data streams are used, one for A, B, and C; a small square block of C is being loaded into L0 cache, A and B reside in cache/memory).

71. Consider claim 6, Gustavson discloses an apparatus, comprising: a memory to store matrix data to be used for processing in a linear algebra program (section 4, line 12, shared main memory and section 4.2, lines 7-9, elements of the matrix); a floating point unit (FPU) to perform said processing (section 3.1, line 4, discloses floating point registers, therefore it is inherent there are floating point units that are doing the multiplications as in section 1, line 2); a load/store unit (LSU) to load data to be processed by said FPU (section 3.1, lines 6-7, load and store operations, thus it is inherent there is a load/store unit), said LSU loading said data into a plurality of floating point registers (FRegs) (section 3.1, line 4, floating point registers); and a cache to store data from said memory and provide said data to said Fregs (section 4.1, line 4, cache), wherein said matrix data in said memory is moved by having inserted moving instructions for said matrix data to be loaded into said cache prior to a need for said data to be loaded by said LSU into said Fregs for said processing, (section 4.1, line 8, algorithmic prefetching), said data being prefetched into said cache from said memory in a nonstandard format (the prefetching is described above in section 4.1, see below for the register block format explanations) predetermined to reduce a number of data streams for a level 3 processing to be three streams (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; note that as only a small square block of C instead of the entire C is being loaded

into the registers, C is essentially a data stream of small square blocks. Also note that streams can be broadly read to be the data from the FPU registers to the FPU itself and thus encompasses A, B, and C regardless of the above technique) and to allow a SIMD (single instruction, multiple data) loading of these streams into said FPU by said LSU (see the second-to-last paragraph of section 3.1, multiple element load instructions), wherein said nonstandard format comprises a register block format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said FRegs (consider a subset or set of matrix data stored in any format in a memory. That matrix data can be arbitrarily split up into blocks of size p-by-q. Regardless of how small or big these blocks of matrix data are, and what data is within these blocks, single or multiple elements of this block of matrix data can be fitted in some way into said FRegs as is necessary for calculations to be subsequently performed), and wherein said three data streams comprise data of one matrix of said level 3 linear algebra processing is considered to be resident in said cache and two remaining matrix operands of said level 3 linear algebra processing reside in a memory or a cache level higher than said cache (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; a small square block of C is being loaded into L0 cache, A and B reside in cache/memory).

72. Consider claim 12, Gustavson discloses for an execution code (section 1, line 6, BLAS code) controlling an operation of said floating point unit (FPU) (section 3.1, line 4, discloses floating point registers, therefore it is inherent there are floating point units that

are doing the multiplications as in section 1, line 2) performing a linear algebra subroutine execution (section 1, line 8, routine along with section 1, line 1, linear algebra), inserting instructions to move data into said cache providing data into said FPU before it was scheduled to be used for processing in said linear algebra subroutine (section 4.1, line 8, algorithmic prefetching), wherein said data is prefetched into said cache from a memory in a nonstandard format (the prefetching is described above in section 4.1, see below for the register block format explanations) to reduce a number of data streams for a level 3 linear algebra processing to be three streams (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; note that as only a small square block of C instead of the entire C is being loaded into the registers, C is essentially a data stream of small square blocks. Also note that streams can be broadly read to be the data from the FPU registers to the FPU itself and thus encompasses A, B, and C regardless of the above technique) and to allow a SIMD loading of these streams into said FPU by said LSUs (section 3.1, first indented paragraph of page 210 as above, number of load and store instructions; there thus must exist multiple loads into said FPU by said LSU. Also see the second-to-last paragraph of section 3.1, multiple element load instructions), wherein said nonstandard format comprises a register block format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said Fregs (consider a subset or set of matrix data stored in any format in a memory. That matrix data can be arbitrarily split up into blocks of size p-by-q. Regardless of how small or big these blocks of matrix data are, and what data is within these blocks, single

or multiple elements of this block of matrix data can be fitted in some way into said FRegs as is necessary for calculations to be subsequently performed), and wherein said three data streams comprise data of one matrix of said level 3 linear algebra processing is considered to be resident in said cache and data for two remaining matrix operands of said level 3 linear algebra processing reside in a memory or a cache level higher than said cache (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; a small square block of C is being loaded into L0 cache, A and B reside in cache/memory).

73. Consider claim 17, Gustavson discloses a method of providing a service involving at least one of solving and applying a scientific/engineering problem, said method comprising at least one of:

using a linear algebra software package that computes one or more matrix subroutines, wherein said linear algebra software package generates an execution code (section 1, line 6, BLAS code) controlling an operation of a floating point unit (FPU) (section 3.1, line 4, discloses floating point registers, therefore it is inherent there are floating point units that are doing the multiplications as in section 1, line 2) performing a linear algebra subroutine execution (section 1, line 8, routine along with section 1, line 1, linear algebra), such that instructions are inserted to move data into a cache providing data for said FPU before it is scheduled to be used in the linear algebra subroutine (section 4.1, line 8, algorithmic prefetching), said data being prefetched from

a memory in a nonstandard format (the prefetching is described above in section 4.1, see below for the register block format explanations) to reduce a number of data streams for a level 3 processing to be three streams (section 3.1, first indented paragraph of page 210 as above, three total data streams are used, one for A, B, and C; note that as only a small square block of C instead of the entire C is being loaded into the registers, C is essentially a data stream of small square blocks. Also note that streams can be broadly read to be the data from the FPU registers to the FPU itself and thus encompasses A, B, and C regardless of the above technique) and to permit a SIMD (single instruction, multiple data) loading of these streams into said FPU (see the second-to-last paragraph of section 3.1, multiple element load instructions), wherein said nonstandard format comprises a register block format wherein data is stored in blocks of size p-by-q where p and q are small integers so that the pieces of these blocks can be fitted into said FRegs (consider a subset or set of matrix data stored in any format in a memory. That matrix data can be arbitrarily split up into blocks of size p-by-q. Regardless of how small or big these blocks of matrix data are, and what data is within these blocks, single or multiple elements of this block of matrix data can be fitted in some way into said FRegs as is necessary for calculations to be subsequently performed), and wherein said three data streams comprise data of one matrix of said level 3 linear algebra processing is considered to be resident in said cache and two remaining matrix operands of said level 3 linear algebra processing reside in a memory or a cache level higher than said cache (section 3.1, first indented paragraph of page

210 as above, three total data streams are used, one for A, B, and C; a small square block of C is being loaded into L0 cache, A and B reside in cache/memory), providing a consultation for solving a scientific/engineering problem using said linear algebra software package (it is inherent that the BLAS will solve some type of scientific/engineering problem for someone who may or may not be the operator of the BLAS program); transmitting a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result; and receiving a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result (it is inherent that the result of the problem will be conveyed to someone who may or may not be the operator of the BLAS program; furthermore, it is inherent that the result can only be shown either through a printout or through some type of electronic means, which encompasses voice through a phone or data through a network that is read via a monitor).

74. Consider claims 2, 11, and 13, Gustavson discloses said timely moving data is accomplished by scheduling move type instructions into time slots existing in a Level 3 Dense Linear Algebra Subroutine. As explained above, it is inherent to prefetching that data is loaded into the cache before the instruction that needs that data is executed, thus there must be a difference between the time of that instruction execution and the time of its data loading, otherwise it would not be prefetching. Furthermore, Gustavson

discloses in page 12, lines 2-3 of section 4.1 that the prefetching instruction does not disturb ongoing computations and data references, thus this prefetching must be done in "time slots" which are independent of other instruction fetching. Gustavson in section 3, line 5, discloses of DGEMM, which is a type of Level 3 Dense Linear Algebra Subroutine.

75. Consider claims 3, 7, and 14, Gustavson discloses said linear algebra subroutine comprises a matrix multiplication operation (section 1, line 2, matrix multiply).

76. Consider claims 4, 8, 15, and 18, Gustavson discloses said matrix subroutine comprises an equivalent of a subroutine from a LAPACK (Linear Algebra PACKage) (section 1, line 1, discloses a BLAS, which is a part of LAPACK).

77. Consider claims 5, 9, 16, and 19, Gustavson discloses said linear algebra subroutine comprises a BLAS Level 3 L1 cache kernel (Abstract, lines 1-6, level 3 BLAS kernel and level 1 cache).

Response to Arguments

78. Applicant argues the double patenting rejection on page 12. However, examiner maintains his rejection and will first reproduce the associated reasoning. Pre-loading, which can refer to the process of loading data into the FPU registers from cache in a

timely manner, must entail loading data *into* the cache in a timely manner as well so *that* that data in the cache can be loaded into the FPU registers in a timely manner. With this interpretation, the teaching of pre-loading must also include some form of prefetching as well. It is noted that the Gustavson prior art would also be able to teach the prefetching limitation as well; however, this is not necessary due to the above interpretation. Preloading or prefetching might mean something more specific in the context of applicant's overall invention and co-pending inventions, and the non-standard format within the co-pending inventions; however, this is not claimed. Examiner is cognizant of the differences between pre-fetching and pre-loading as implied by the associated claimed limitations, but the pre-loading can nevertheless necessitate prefetching as well (which does not mean that they are the same).

Applicant first states that a terminal disclaimer would be moot merely because the co-pending applications were filed on the same day; however, as explained, a terminal disclaimer also prevents separate ownership of the co-pending applications.

Applicant again argues that the preloading in the '937 application is an alternative to the prefetching method of the present invention; however, this is irrelevant to the issue of double patenting. MPEP 804 states: A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). In determining whether a nonstatutory basis exists for a double patenting rejection, the first question to be asked is — does any

claim in the application define an invention that is merely an obvious variation of an invention claimed in the patent? If the answer is yes, then an "obviousness-type" nonstatutory double patenting rejection may be appropriate.

In other words, although the preloading described in the '937 application may be an alternative to the prefetching method of the present invention, as conveyed in the specification of one or both of the applications, this does not change the fact that *the claims* of the instant application are an obvious variant of *the claims* of the '937 application. As explained above, it is apparent that pre-loading, which can refer to the process of loading data into the FPU registers from cache in a timely manner, must entail loading data *into* the cache in a timely manner as well so *that* that data in the cache can be loaded into the FPU registers in a timely manner.

Applicant argues that the preloading is an alternative method to overcome a one or more cycle penalty associated with the cache/FPU loading of the newer machines. However, this does not appear in either the instant claims or the claims of the '937 application. Applicant argues that preloading entails the rearrangement of incorrectly loaded data to be in a correct format. This too does not appear in either the instant claims or the claims of the '937 application. As the issue of double patenting is directed toward the claims, the fact that the specification of the '937 application may disclose that preloading is an alternative to prefetching or that preloading rearranges incorrectly loaded data is irrelevant. As explained above and in previous actions, examiner is cognizant that there are differences between the examiners interpretation of prefetching and preloading, and the role of prefetching and preloading in the applicant's overall

invention; however, these differences are not explicitly recited in the claims, which allow for the examiner's interpretation.

79. Applicant argues on page 13 that section 3.1 of Gustavson does not make any suggestion whatsoever about the format used for prefetching. However, the examiner was validly broadly interpreting the claimed limitation "said data being prefetched into said cache from said memory in a nonstandard format" to mean that the manner of prefetching is done in a nonstandard format and not the data itself, which does not entail hindsight.

80. Applicant argues on page 14 that pre-SIMD machine were not capable of reducing the data streams down to only three streams. However, Gustavson nevertheless teaches the claimed limitation as explained in the rejection above. It may be the case that a broad interpretation of the limitation "data streams" is the reason why Gustavson still teaches the claimed limitation, but the citation is nevertheless valid.

81. Applicant again argues on page 14 that prefetching and preloading are alternative methods; however, this does not change the fact that the instant *claims* are obvious variants of the *claims* of the '937 application, as the prefetching and preloading limitations can be broadly interpreted despite any disclosure in the specification relating to their context which state that they are alternatives or involve the rearrangement of data.

82. Applicant argues on page 15 that the prior art refers only to multiple loads of load multiple type $k=1$, whereas the present application addresses architectures capable of a SIMD load with $k > 1$. However, Gustavson discloses of multiple element load instructions, which appears to meet the SIMD limitation. If the SIMD architecture in the instant application is further different from the prior art reference, those differences should be claimed.

83. Applicant argues on page 16 that the examiner's citation of data being prefetched into said cache from said memory in a nonstandard format has nothing to do with data format; however, the claim does not necessitate that it is the data and not the prefetching which is in a nonstandard format.

84. Applicant has amended in the details of a register block format. However, applicant's amended specification appears to disclose that a row- or column-major format is an example of a register block format, in which case the prior art would teach the limitation as applicant has previously argued that the prior art was in a standard row- or column-major format. Additionally, the claimed description of a register block format appears to be sufficiently generic such that the prior art, regardless of what format is used to store matrix data, would teach the claimed limitations. Examiner recommends elaborating on this register block format to overcome these two positions.

Conclusion

85. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Keith Vicary whose telephone number is (571)270-1314. The examiner can normally be reached on Monday - Thursday, 6:15 a.m. - 5:45 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183

/Keith Vicary/
Examiner, Art Unit 2183